**1. Project Scope**

**Objective:**

To develop a **secure and efficient Banking Application** that enables:

* Users to **create accounts, check balances, and transfer money**.
* An **admin to manage user accounts** (create, delete, credit/debit funds, reset passwords).

**Target Users:**

* **End-users:** Customers who need online banking services.
* **Admin:** A bank administrator who manages user accounts.

**2. Requirement Analysis**

**Functional Requirements:**

1. **User Functionalities:**
   * User registration with **username, password, email, mobile number**.
   * Login with authentication (username/password).
   * View account details (account number, balance).
   * Transfer money to another account.
   * Receive money and view transaction history.
2. **Admin Functionalities:**
   * Login with admin credentials.
   * Create and delete user accounts.
   * Assign account numbers to users.
   * Credit/Debit funds from user accounts.
   * Reset user passwords.

**Non-Functional Requirements:**

* **Security:** Password encryption, authentication validation.
* **Performance:** Fast execution of banking transactions.
* **Scalability:** Handle multiple users and transactions.
* **Usability:** User-friendly interface with clear instructions.

**3. System Design**

**Architecture:**

* **Frontend:** C-based CLI (Command Line Interface).
* **Backend:** File-based storage (or database if extended).
* **Data Storage:** Structured using **C structs and file handling**.

**Database Schema:**

| **Table** | **Fields** |
| --- | --- |
| Users | ID, Username, Password, Mobile, Email, Account Number, Balance |
| Transactions | Transaction ID, Sender ID, Receiver ID, Amount, Date |

**Use Case Diagram:**

* **User**: Login → View Account → Transfer Money
* **Admin**: Login → Create User → Manage Accounts → Logout

**4. Development Plan**

| **Stage** | **Tasks** | **Timeline** |
| --- | --- | --- |
| **Requirement Analysis** | Gather user/admin needs, define scope | **Day 1** |
| **System Design** | Design database schema, flowcharts | **Day 2** |
| **Development** | Implement login, transactions, admin panel | **Day 3-4** |
| **Testing** | Test cases, bug fixes | **Day 5** |
| **Deployment** | Host and optimize code | **Day 6** |
| **Documentation** | Create user/admin guides | **Day 7** |

**5. Risks and Mitigation**

| **Risk** | **Mitigation Strategy** |
| --- | --- |
| Security issues | Implement password encryption |
| Data loss | Regular backups for file/database |
| Performance lag | Optimize data structures and transactions |